MATERIALS WITHIN BLENDER
By Olivier Saraja

LinuxGraphic.org

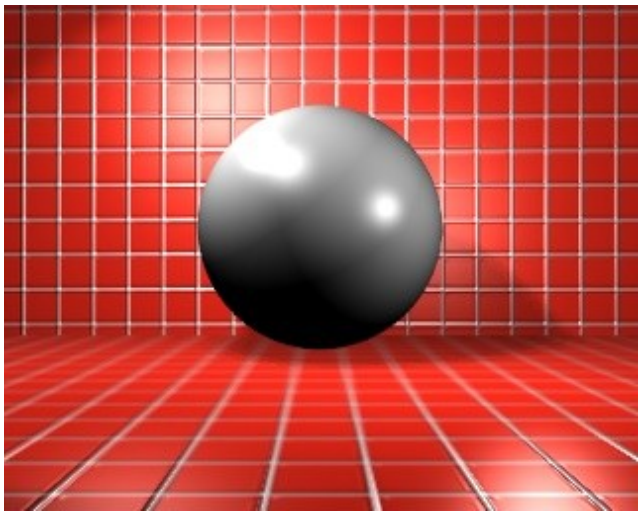
Introduction – **Color** – **Speculars** – **Mapping** – **Channels**

Introduction

One of the keys for the rendering of a good scene relies on the appropriate choice of the material properties of objects within it. You can have the most beautiful, the most original and the finest models of the world, poorly managed materials can nothing but spoil your scene.

It is important to understand how work the many Material Buttons in order to use them the best possible way.

To help us in this understanding task, I will introduce the basic 3D artist's tool : the sphere ! It will help us visualize the modifications induced by the different material options, and thus, to understand them better.

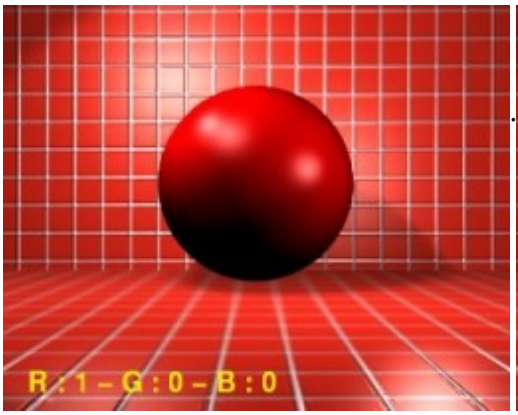


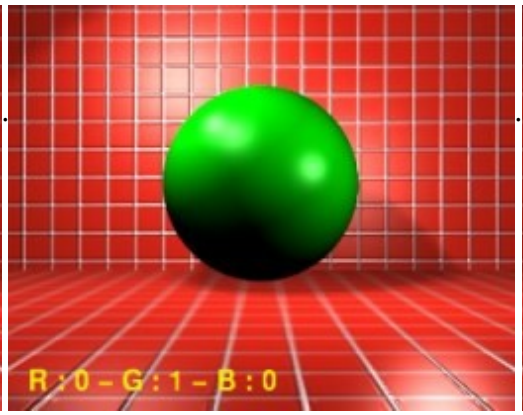
The Color :

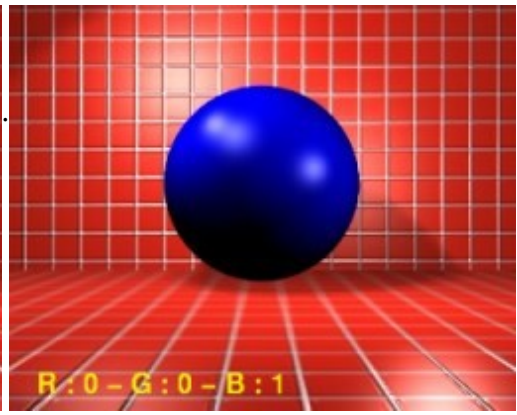
It is defined and altered by the sliders showed below.

Red, Green and Blue

R, **G** and **B** defines the three Blender's standard color channels (respectively Red, Green and Blue). The default value for each of these three colors is 0.8, which results in a light grey. You should feel free to play with each of these sliders until you get, in the previsualization little window, the color you are looking for. Here are, for example, the results for the three fundamental colors once rendered with every other material parameter set to its default.







It is important to note at this stage that if you use a picture as a texture map to your sphere, it is the R, G and B components from the picture that will be used to color your sphere, according to mapping rules we will talk about later. But even if you use a texture to color your sphere, it is highly recommended to give your material a R, G, B color for the following points :

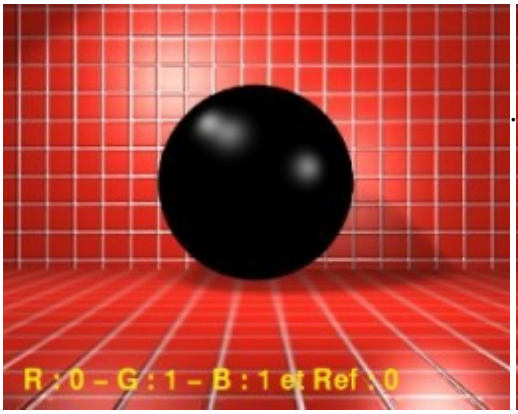
- in solid preview (ALT+Z) of your scene, you can more easily judge how the objects interacts with others if they are of different colors ;
- while using the Radiosity module, it's the fundamental R, G, B color that is taken in

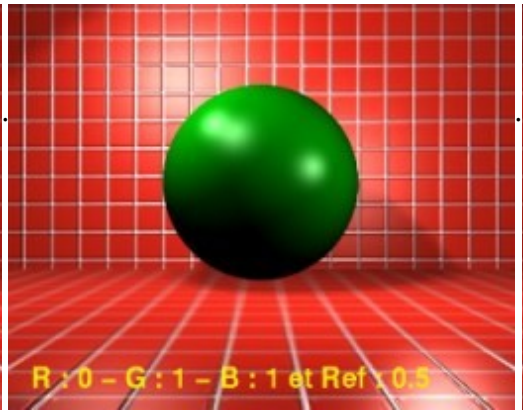
account by the radiosity engine, and not the components given by the mapped texture ;

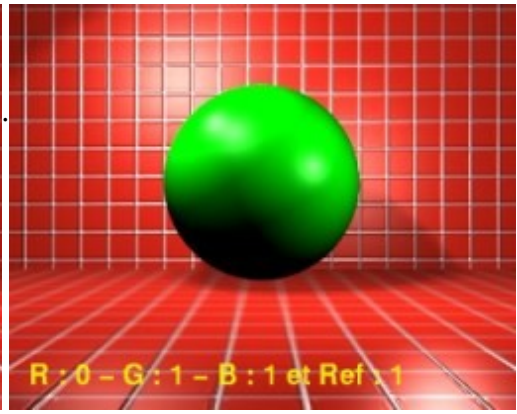
- while rendering intermediary environment maps (EnvMap), it is useful to give an object a color very close to the one of the same object with its final texture, because of the limitations of this kind of maps within Blender (the objects with an EnvMap texture don't reflect each other, so the reflects show flaws as annoying as the lack of objects on the map ; we will discuss later how to painlessly bypass this limitation).

Reflection

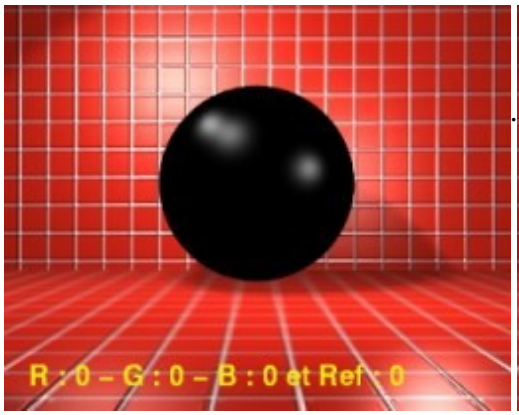
This parameter (**Ref**) defines how much light is returned by the material. To a given **R, G, B** color, a low **Ref** value gives a dark color, and a high one a light color. Thus, we can easily obtain the following colors :

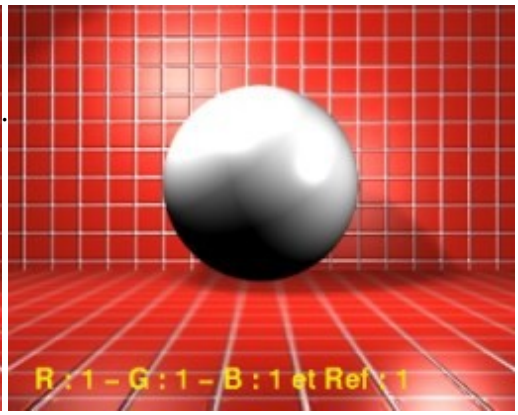






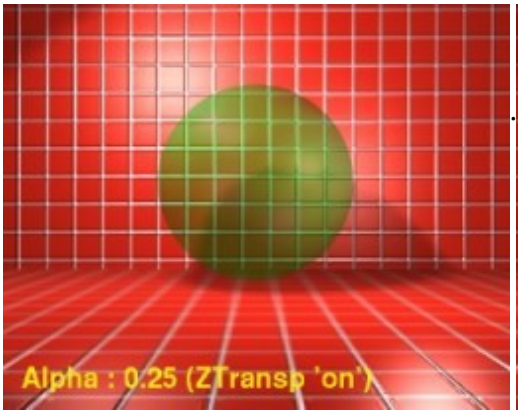
Here we come with a useful property to know : to achieve a true black ball (R:0, G:0 and B:0), you should set **Ref** to 0. Of course, to achieve a true white ball, (R:1, G:1, B:1), **Ref** should be set to 1. The later case is particularly useful when you try to model a white paper sheet or a snowy landscape and, without taking this fact into account, you will get, at best, a pale grey instead of a pure white.

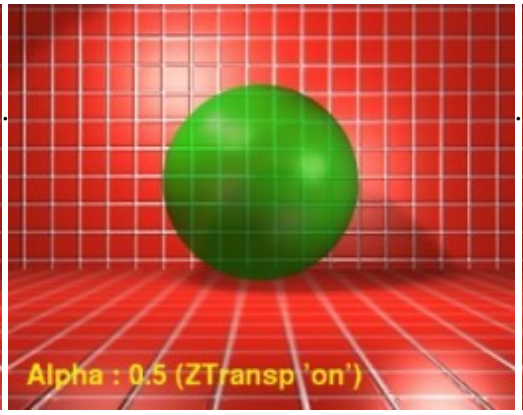


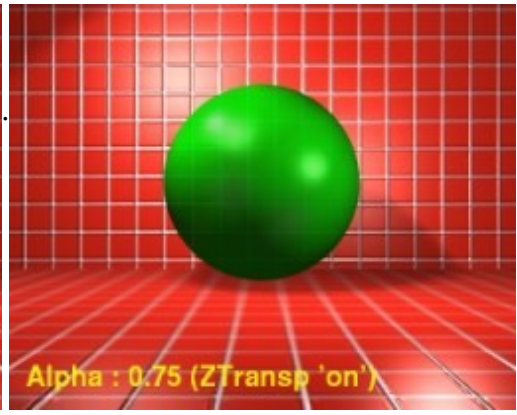


Alpha

This parameter (**Alpha**) defines the color's opacity, e.g. how much the color of the objects located behind will be seen through the material. The best results are achieved when the "**ZTransp**" button is activated.

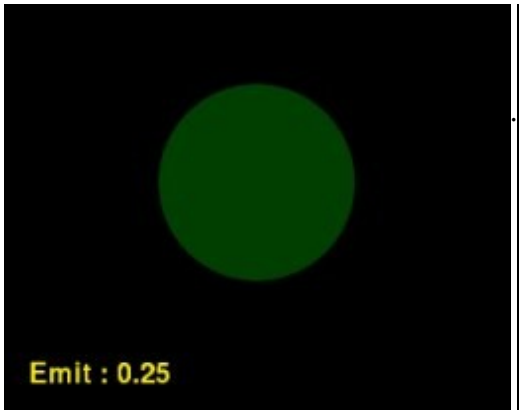


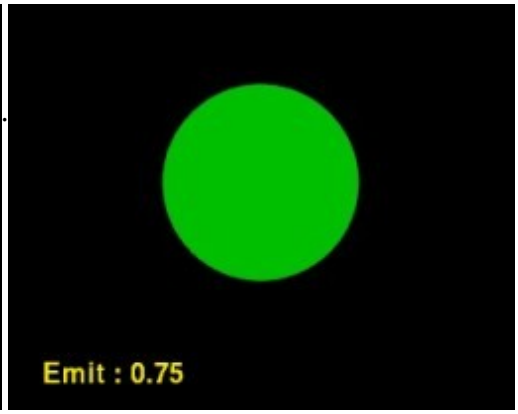




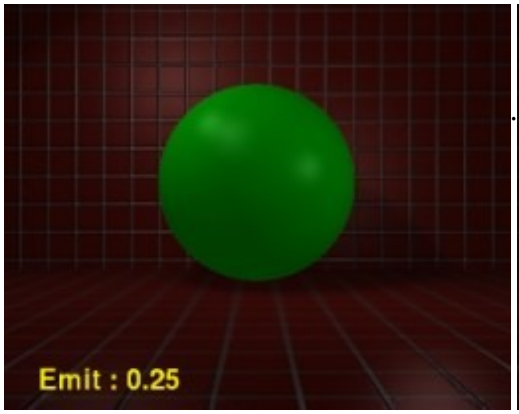
Emittence

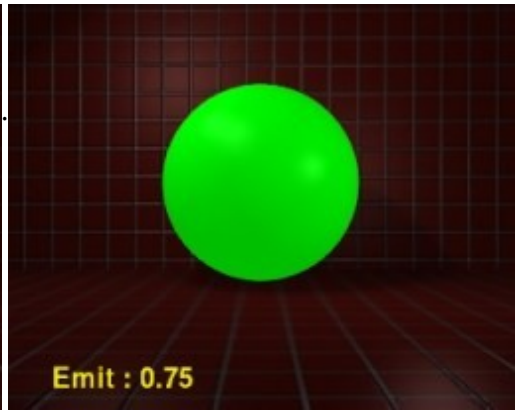
This parameter (**Emit**) simulates a luminescent material, e.g. a material that can shine in the dark. This means that in absolute darkness, our material, with a **Emit : 0**, is fully invisible, while with a **Emit : 1** il will be visible as well as in full daylight ! The two following pictures have been rendered without any light sources.





The two same pictures, but with a slight lighting (the **Energies** of the lamps and spots have been set to 0.2).

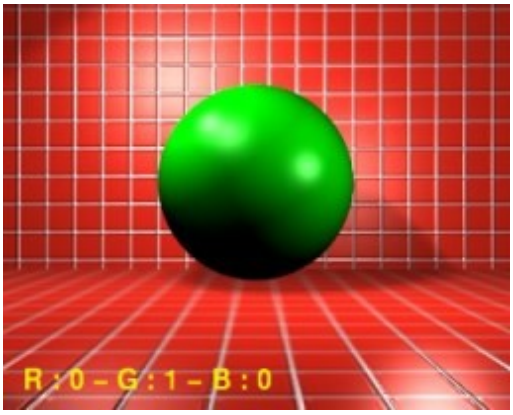




You should note that this parameter is not sufficient to cast light on other objects ! This is possible only through the use of the Radiosity engine which is the only actual way to take into account the luminosity emitted by objects other than lamps in Blender.

S p e c u l a r s :

The light color patches we previously saw on the sphere are called speculars :



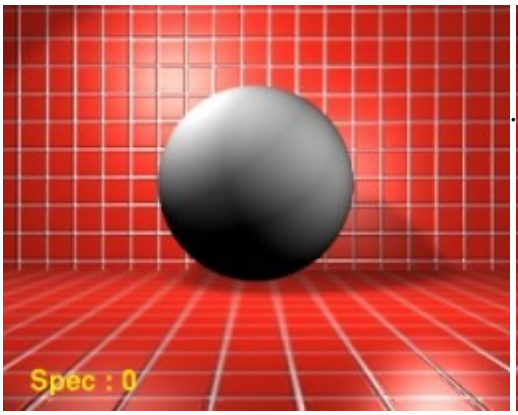
The sliders from this block of parameters will help us to define the size, the blurriness of the borders and of course the color of these speculars. Thus, we can create the illusion of a bright or varnished material, or, on the opposite side, a dull or dirty material, according to our needs.

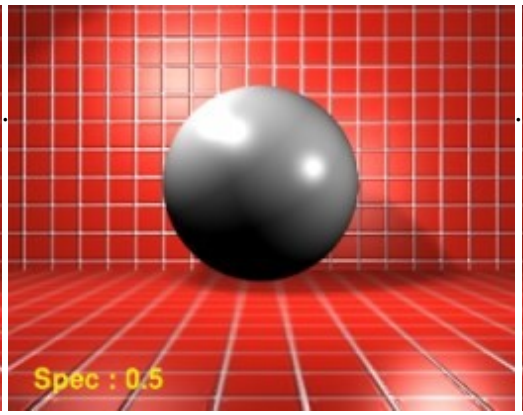
Specular color

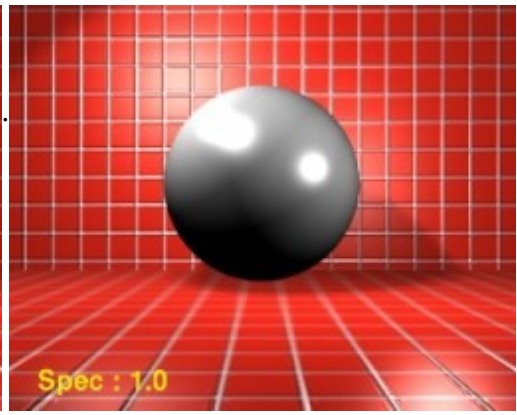
First of all, select the **Spec** button. The R, G and B sliders we studied in the previous section now give us control over the color of the speculars. In this tutorial, we will leave the speculars to their white default color, but it is often useful to give a speculars' material a slightly lighter color than the base color.

Specularity

The **Spec** slider defines the intensity of the speculars returned by the material. Varnished, metallic or wet surfaces generally tend toward a high **Spec** value, while dirty, dull ou dusty materials tend to have a low or null **Spec** value. The following pictures should illustrate this, but remember that all the other parameters have been left to their Blender's default. It's interesting to note that you can set **Spec** to a value as high as 2.0, which is particularly interesting for materials supposed to simulate glass.

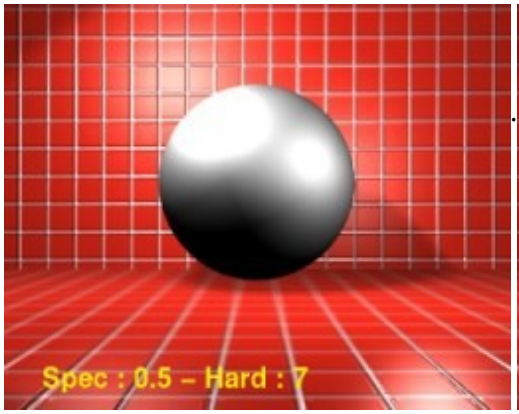


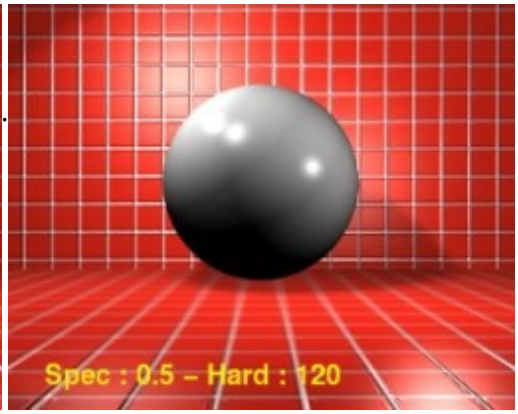




Hard

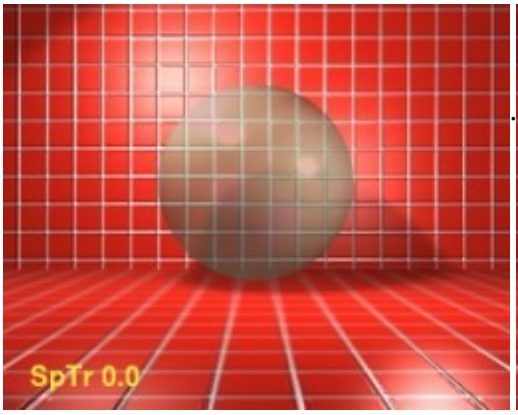
This slider defines the hardness (or the sharpness) of the specular's frontier. A low value describes a blurry specular's border, with large color diffusion, and is appropriate to plastic, skin or a surface reacting almost uniformly to light. A high value shows sharp borders, just like metals, ceramic or a wet object.

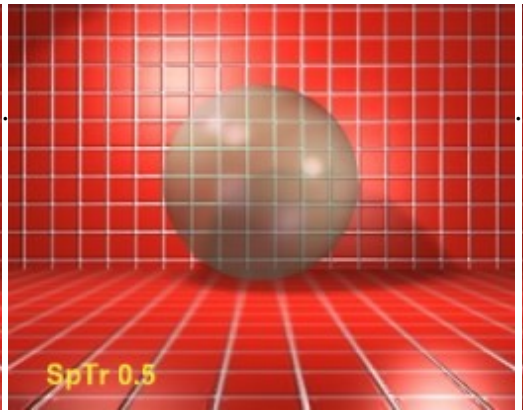


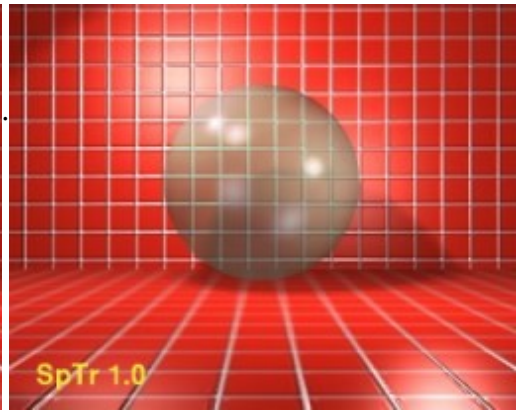


Specular Transparency

This slider (**SpTr**) is to be used along with the **Alpha** and **ZTransp** material properties. It allows to decide the opacity of the specular area and to enhance the rendering results for material whose purpose is to simulate glass. The higher the SpTr value, the more the specular area is opaque and reflects light.

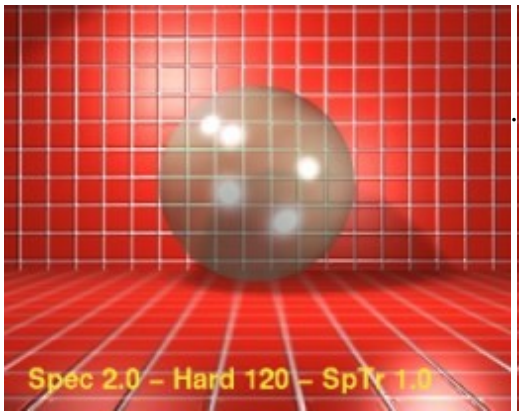


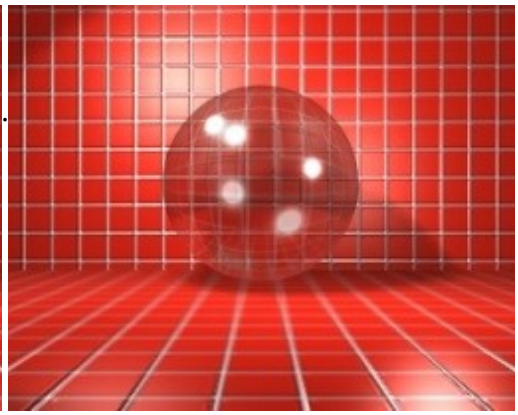




Spec 0.8 − **Hard** 120 − **Alpha** 0.35 − **ZTransp** 'on'

With a saturated **Spec** value, and a **EnvMap** texture type (**CMir** activated), we can achieve this cool glass effect.





However, we are far from reaching a satisfying realism with this pseudo−glass ball rendering. This is why we offer you the following practice.

P r a c t i c e s :

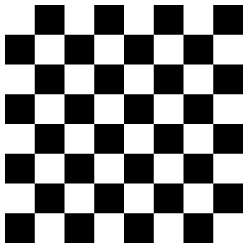
Modelling a Glass
Material Indices

Textures Mapping:

Textures are, fundamentally, 2D pictures, and the objects of our scenes 3D entities. Predictably, we always have to think about the good mapping we must use with our object in order to make the texture looking right. This chapter shows how works mapping within Blender.

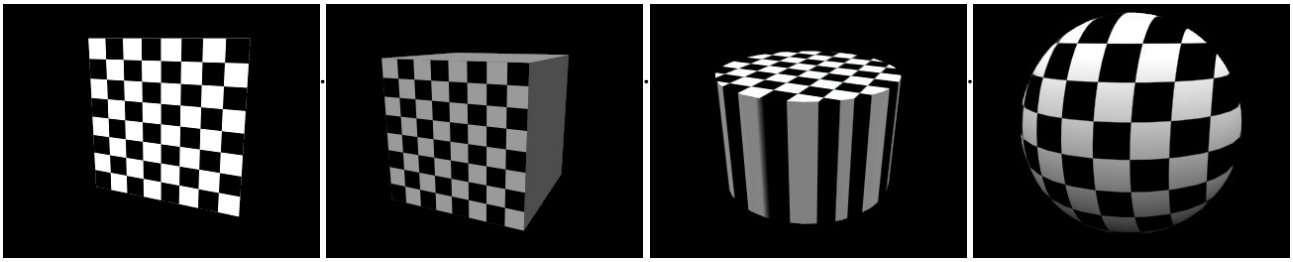
In fact, in the Material Buttons (**F5−KEY**), Blender lets you make use of four types of mapping, each one shown in a separate button. We will use each of these options with the corresponding primitives, in order to understand what the best usage of these mapping settings should be.

For a start, let's get a clear (and very visual) texture.



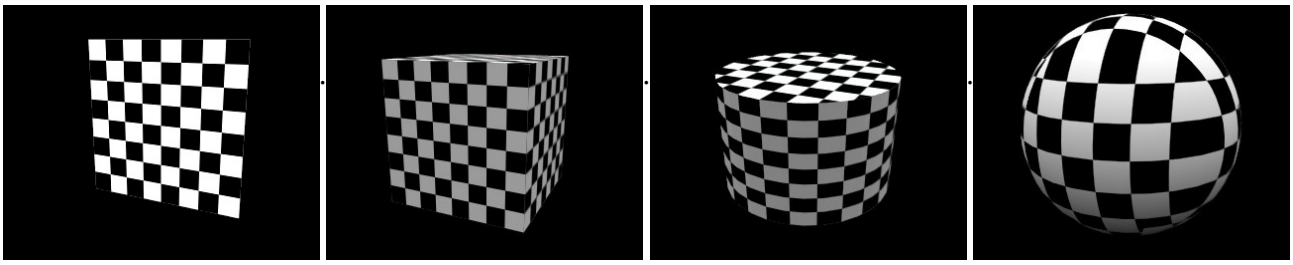
In this demonstration, I created four primitives issued from Blender : a plane, a cube, a cylinder and a sphere. For each of these, I made a render after having toggled from a mapping option to another. The pictures below are self explanatory.

Flat



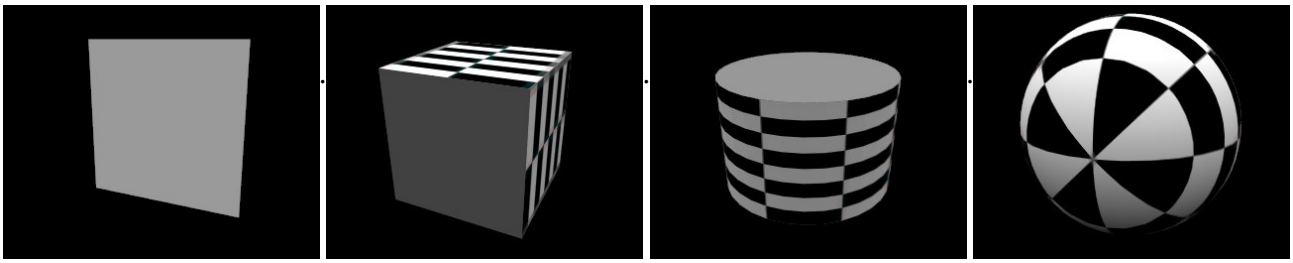
If we want to understand how the **Flat** option maps the textures, we should figure how works a cine−projector on the screen... We should also note that the mapped texture is also visible from the 'back' of the object, e.g., the texture is seen 'through' the object.

Cube



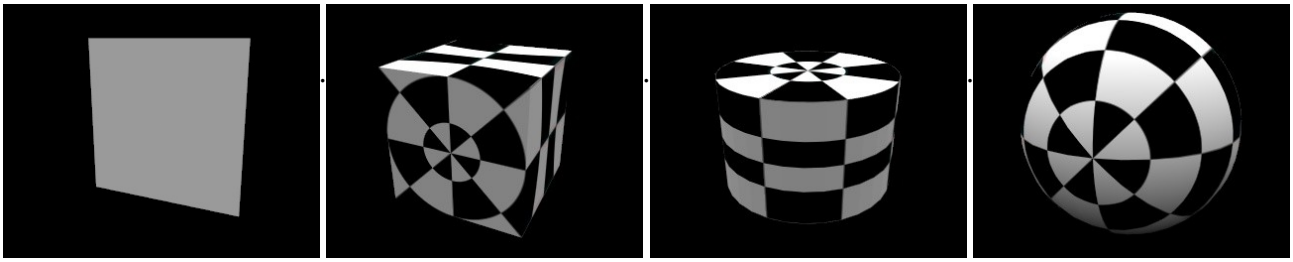
The **Cube** option forces the display of the texture in the six directions. This is why the cylinder can show texture on its curved part, and why the cylinder shows a texture discontinuity.

Tube



The **Tube** option can be compared to a paper sheet rolled in the direction of its wideness, with both ends perfectly joining. The results on the plane and cube primitives are interesting to note : the virtual paper sheet is rolled around the front plane surface.

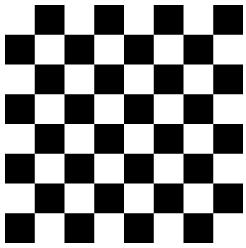
Sphere



The **Sphere** option works the same way as the previous option, but this time, the paper sheet is rolled both in the wideness and in the length directions. Again, the results on the plane and cube primitives are interesting to note.

Textures channels :

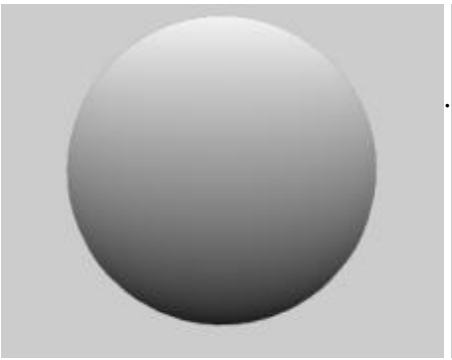
It is also possible, with Blender, to specify which material property will be modified by a given texture. Let's take back our previous texture example, which will illustrate our purpose.



The texture channels can be individually set using the group buttons appearing on the far right end of the Material Buttons (**F5**).

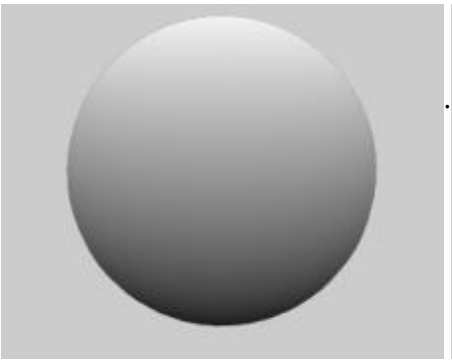
It is very useful to rename each texture in order that each name becomes self−explanatory. In the previous picture, for example, we can see all the eight **channels**, appearing as grey buttons, hosting the textures named "Chan1", "Chan2", "Chan3" and "...".
The picture hosted by each of these channels can be redirected to one of the following **outputs**, appearing as blue−green buttons : **Col, Nor, Csp, Cmir, Ref, Spec, Hard, Alpha, Emit**. Its color intensity (or shade of grey intensity) is then used in order to affect the base material, either by mixing with it (**Mix**), by multiplying its intensity by its own (**Mul**), by adding its intensity to its own (**Add**) or by substracting its intensity to its own (**Sub**). In the following tutorial, we will only pay a look to the results of the Mix button. You should get various interesting results if you try the other buttons (Mul, Add and Sub).

The Col output
This is the default output when you add a texture to a material, and the most useful of all. The color datum of the texture is applied to the material. The **Col 1.000** slider lets you set the color intensity of the channel. You can easily figure this out by making some renders with this slider set to 1.000, 0.000 or 0.500, for example. The following pictures show the raw base material, the the same material with our study texture redirected to the **Col** output.



The Nor output

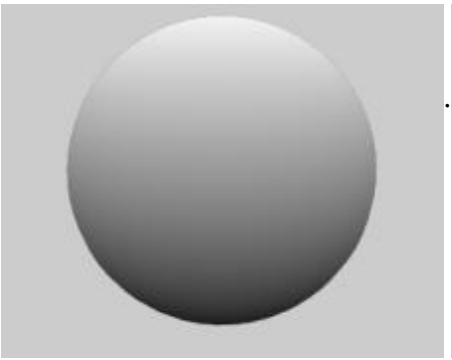
Another very useful output, also with its own slider : **Nor** 0.500. The color datum of the texture is passed to the material as a height map for the surface of the object. To be perfectly clear : dark areas of the texture are computed as hollow and the light areas as bumps. Please note that the **Nor** button has two positions, the second appearing with yellow text, and for which the data are inverted : the dark areas appear as bumps, and resp. The following pictures show the raw base material, then the same material with our study texture redirected to the **Nor** output, and then to the **negative Nor** (in yellow). The **Nor** 0.500 slider lets you set the intensity of the change applied by the texture to the **Nor** output, enabling you to get more or less significant bump maps based on the texture. The **Nor output** is the default output associated to the procedural texture '**Stucci**'.



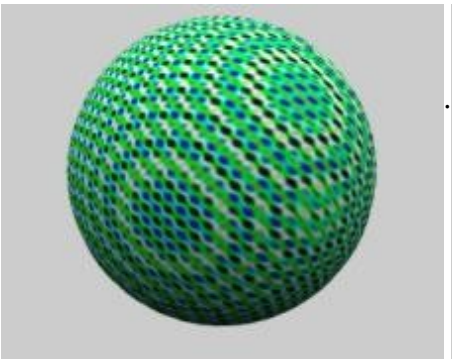
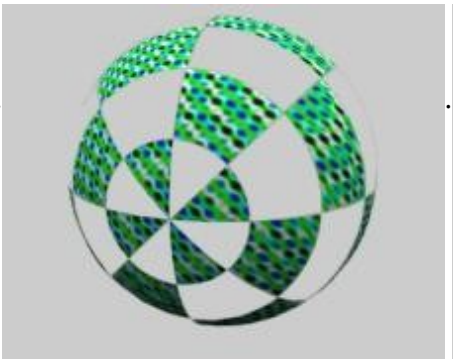
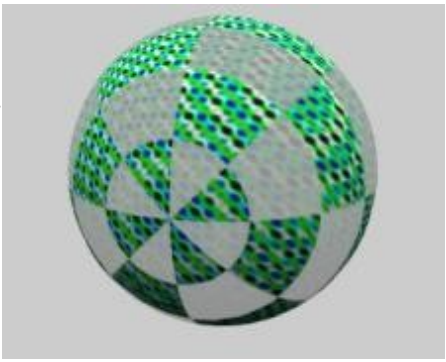
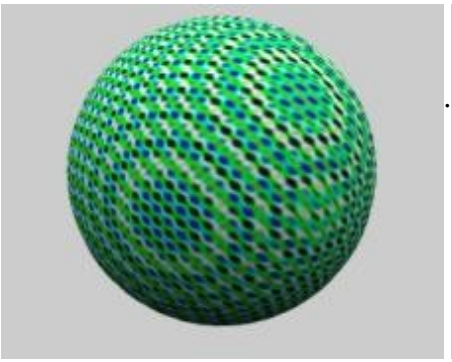
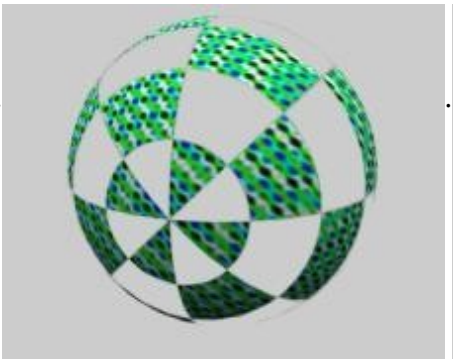
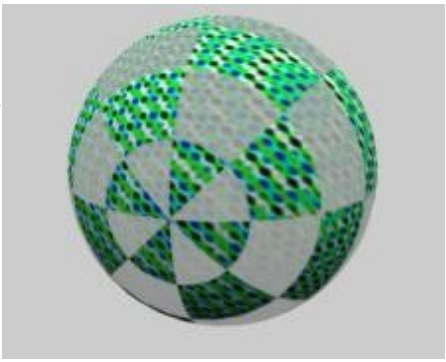
The other outputs are slightly more specific, but can easily find many interesting uses in your renderings. Their intensity is set by the **Var 1.000** slider the same way the **Col 1.000** and **Nor 0.500 sliders** affect the **Col and Nor** outputs. These two last outputs being the most useful, they each have their own slider.

The Ref output

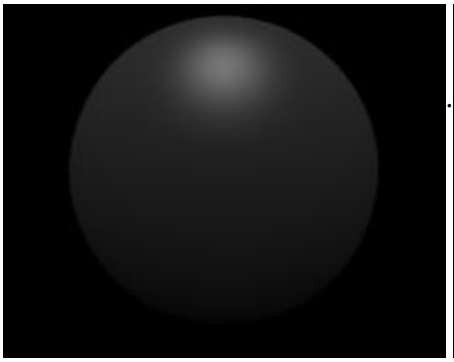
With it you can pass the texture intensity datum to the **Reflection** material property, and thus tinge the lightness of the base material. This output admits an inverted setup (text in yellow).



The Alpha output
In order to take advantage from this output, you should turn on the **UseAlpha** option (and even the **CalcAlpha** and **NegAlpha** if the picture is not a Targa file coded on rgba keys) in the Texture Buttons (**F6**). You also need the **Alpha** parameter to be set to a value different from 1.000 in the Material Buttons (**F5**). The following pictures feature the raw material ((coloured in order to have a better view of it), then the same material with our study texture redirected to the **Alpha** output (with the parameters set to **Alpha 0.000** and **Alpha 0.500** on the second and third pictures). This output admits an inverted setup, as featured on the fifth and sixth pictures, even if this effect can be achieved through the use of the **NegAlpha** button. Of course, the inverted **Alpha output** and the **NegAlpha** button invert each other.

The Emit output
This output allows you to pass the texture intensity datum to the **Emit** material property. This output admits an inverted setup (text in yellow). The following pictures depicts the raw base material lighted by a low−energy Hemi lamp, then the same material (with same lighting conditions) with our study texture redirected to the **Emit output** with **Dvar** set to 0.500, and the last picture the same texture redirected to the **Emit ouptut** with **Dvart** set to 1.000.



The Csp, Spec and Hard outputs
At a final point, the effects of the **Csp**, **Spec** and **Hard** outputs are easily figured out with high Spec and Hard material properties, in various setups too toilsome to present here. As they are the less useful outputs, I won't write about them, but they could be of real use in modellings glass objects. **Csp** lets you play with the specular color, **Spec** with the size of the specular and **Hard** on its fuzziness.

The Cmir output
This output is obsolete and won't be documentated here.

Some Hints:

- The same texture can be redirected simultaneously to one or many different outputs. For example, your **Col** texture shows dark and light areas than can be playful with the **Nor** output.
- Use (each time you can) textures with transparent backgrounds (something like stencils or masks) by activating the **Alpha** output (read above on this page about the other buttons to activate in order to take advantage from this option) along with the output you want to use. You will get more predictable results when you use the Mix, Mul, Add or Sub option buttons.

Even if the same texture could be useful to the **Col**, **Alpha** and **Emit** outputs at once, for example, try to put a copy of the texture in a different channel and redirected to an unique output at a time. According to the previous example, the **Col** slider lets you choose how mucj the texture color will affect your object through the **Col** ouput, but the **Dvar** slider will act over both the **Alpha** and **Emit** outputs at a time if these three outputs are activated at the same time for the same texture. Most of time, you would like to act upon the **Alpha** output independently from the **Emit** output and override the existence of only one **Dvar** slider per channel.